RESEARCH NOTES

September 2007

Distributed Paramics Development
(ATMS TESTBED PHASE III FINAL REPORT)

Why was this Research Undertaken?

Micro-simulation modeling is an increasingly popular and effective tool for analyzing a wide variety of dynamical problems, which are not amendable to study by other means. As a suite of ITS-capable, user-programmable, high-performance microscopic traffic simulation package, PARAMICS offers very plausible detailed modeling for many components of an 'ideal' simulator. It has become a widely used microscopic simulation model in the US (especially in California). However, for it to be incorporated effectively for real-time applications involving California's urban freeway networks, it needs to have scalability, so that these large networks can be simulated in something close to (or, even exceeding) real time.

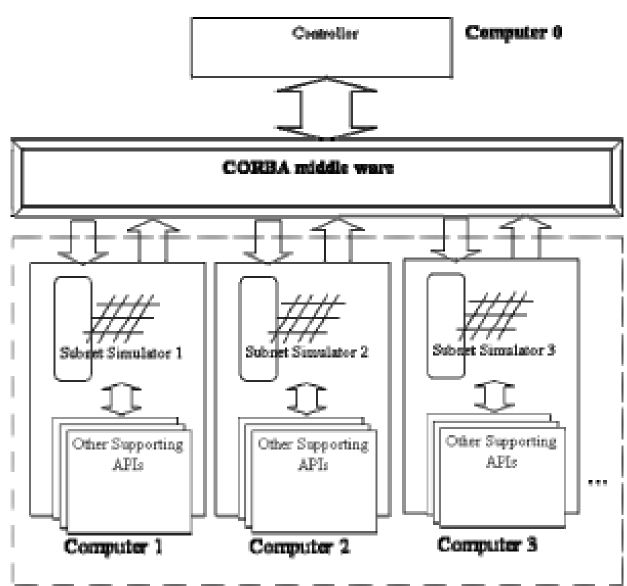
For more in depth discussion and technical analysis, refer to TTR3-07 (Testbed Technical Report).

What was done?

Research is being conducted under the auspices of both PATH and the Testbed to develop a distributed version of Paramics that will be truly scalable. In this project, we developed a distributed modeling framework for large-scale microscopic traffic simulation and implemented a scheme with global control and independent subnets using the PARAMICS model. Unlike the previous studies using the dedicated high performance machines, our efforts are to utilize the low-cost networked PCs that are commonly available. By using the Application Programming Interface (API) functions supported by off-the-shelf PARAMICS, we are able to simulate the traffic in a large region simultaneously with independent subnets running on separate desktop PCs and vehicle transferring from one subnet to another.

In the distributed simulation environment being developed, the targeted large network is divided into sub-networks, and each sub-network is simulated on a separate desktop PC. The general distributed architecture includes: 1) a "controller" simulator running the "master network," and 2) several sub-network simulators. Although the controller may have various tasks related to coordinating the traffic simulation itself, the essential task from a computational architecture standpoint is the synchronization of the time in each sub-network, either at every simulation time-step or at specified time intervals. To synchronize the simulation time, the controller has the ability to start and stop the sub-network simulation at any time. In addition, such information as boundary zones and their corresponding ownerships are established in the controller computer.

During a simulation run, the controller and simulators communicate over the distributed platform. The sub-network simulators act as slaves to the controller. During a time step of simulation or certain time interval, a simulator executes a non-blocking loop (asynchronous communication) while waiting for a new request from the controller. A request is simply a message associated with a specific task. When the request arrives into a sub-network simulator, it starts with an execution of the corresponding sequential code. When the request task is completed, a notification is sent back to the controller. When all simulators are "checked in", the simulation master clock advances by until it reaches the master clock time. The communication between the simulators and controller is through the CORBA distributed platform. A pictorial description of the scheme used for distributed processing is shown in the figure below.



Distributed Paramics Schematic

What can be concluded from the Research?

Performance testing and analysis of the implemented prototype demonstrate that the proposed framework is very promising. Since synchronous communication is used among simulators and controller, each simulator can only run as fast as the slowest one; proper and balanced decomposition of the network is critical to the overall performance. Because the total computational requirement for a microscopic traffic simulation is dominated by the number of vehicles in the network at any time, the ideal division of network is to create N regions that each has exactly V/N vehicles, where V is total number of vehicles in the simulation and N is the target number of processors. The speed-up performance of the simulation in distributed processing is also dependent on the communication to computation overhead: if there are a large number of communication operations for each computational operation, the overall process will reduce in speed. In order to minimize the communication to computation overhead, distributed simulations require methodological decomposition of the large network to find a subdivision where there are as few boundaries as possible and the computational load is spread evenly across the processors.

In the design, not only does the controller synchronize the time clock of simulators but it also manages the global abstract network, global O-D matrix and global routing table. The benefit from this design is that vehicle's origin-destination and its path are all controlled at the global level, as opposed to the local level. In this aspect, the simulator's design is similar to the simulation over single processors in terms of routing, with the distinction of updating vehicle's location over distributed processors. But the communication load between the controller and simulators is also significantly higher than that of the light controller-heavy simulator, which may slow down the simulation.

The controller has the global abstract network, global O-D matrix and global routing table. Each simulator also has its own local routing table. When a vehicle is generated in the sub-network, if its origin and destination belong to different sub-network, its temporary destination in the originating sub-network will be determined from the global routing table, but its path in the originating sub-network will be determined locally from the local routing table. Instead of routing every individual vehicle at the global level, the design allows a vehicle's route calculated at the local level, and significant communication overhead is reduced

What do the Researchers Recommend?

It is recommended that the modeling platform be tested and on several large networks. For early testing, it is recommended that the large Orange County Testbed network recently fine-tuned at UCI be used. This network is one of the largest networks coded in Paramics, with as many as 100,000 vehicles being present at any given time.

Implementation Strategies

The research demonstrated in this paper is a light global control and independent subnets design. To achieve a more accurate distributed simulation results, the heavy global controller and coordinated subnets design should be considered in the future research projects, including development of the algorithm for global routing, and network decomposition.

List of Contacts

Principle Investigator:
Dr. Will Recker
Institute of Transportation Studies
University of California; Irvine, CA 92697-3600
Tel.: (949) 824-5642: Fax: (949) 824-8385
Email Address: wwrecker@uci.edu

Other Investigators:
Dr. R. Jayakrishnan
Institute of Transportation Studies
University of California, Irvine
Irvine, CA 92697-3600
Tel.: (949) 824-6848; Fax: (949) 824-8385
Email Address: rjayakri@uci.edu

Contract Manager:
Fred Yazdan
Caltrans, Division of Research and Innovations
Office of Technology Applications
Advanced Testbed Program Branch
6681 Marine Way, Irvine, CA. 92618
Tel.: (949) 936-3462; Fax: (949) 936-3461
Email Address: fred.yazdan@dot.ca.gov